

Digitalizador 3D

Di Nardo, Juan Diego
juandiegodinardo@yahoo.com.ar

Insua, Leandro Martín
leandro_ins@yahoo.com.ar

Toth, Gastón Alejandro
gastontoth@yahoo.com

Universidad Nacional del Comahue
Facultad de Economía y Administración
Departamento de Ciencias de la Computación

Resumen

En este trabajo presentamos una idea práctica para digitalizar objetos en tres dimensiones. Para lograr la digitalización construimos un dispositivo con componentes básicos y económicos. Este dispositivo se conecta al puerto de joystick disponible en cualquier computadora personal por lo cual los requerimientos de hardware son mínimos. Como complemento del digitalizador programamos un conjunto de aplicaciones y herramientas que permiten obtener los puntos del objeto, hacer una representación digital, visualizarlo y modificarlo.

Introducción

La motivación para llevar a cabo este proyecto es lograr la digitalización de objetos mediante una interfase hardware utilizando recursos mínimos, permitiéndonos, una vez obtenida la representación digital del objeto, visualizarlo y modificarlo usando herramientas de software.

Frente a la complejidad y costos de la tecnología existente, optamos por una tecnología simple y accesible.

Dividimos este sistema en tres partes principales:

- El periférico con el cual se toman los puntos del objeto.
- El programa que comunica al dispositivo con la computadora y transforma los datos en su representación digital.
- La aplicación que permite la visualización y modificación de los objetos.

La vinculación entre el programa que maneja al dispositivo y la aplicación es la estructura del archivo obtenido en la digitalización. Esto brinda flexibilidad para cambiar el periférico por otro de mejor tecnología y con mayor precisión. Esta independencia también facilita la agregación de nuevas funcionalidades al programa de visualización.



Digitalizador

El digitalizador es el dispositivo que indica la ubicación de los puntos de un objeto en el espacio. La tecnología que utilizamos para la obtención de los datos es la del puerto de joystick, presente en la mayoría de las computadoras personales.

El digitalizador consta de tres segmentos móviles ubicados en forma consecutiva, los cuales permiten la movilidad necesaria para obtener los puntos que conforman la estructura básica del

objeto. En el último segmento hay un pulsador que, cuando lo ponemos en contacto con el objeto, envía una señal al programa para que inserte ese punto al conjunto de puntos anteriormente obtenidos. El último de los segmentos está sujeto a una columna fija sobre la base. Dicha base contiene además, un disco giratorio sobre el cual colocamos el objeto.

En cada uno de los ejes que hacen posible la movilidad de los segmentos agregamos un potenciómetro (resistencia variable) que indica los grados que se mueve un segmento con respecto al anterior.

La base circular giratoria en la cual situamos el objeto está dividida en 72 dientes con un ángulo de 5 grados entre cada uno de ellos. Esto permite obtener 72 perfiles del objeto, completando los 360 grados y cubriendo así todo el objeto.

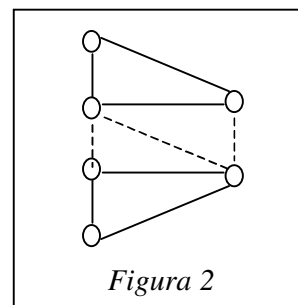
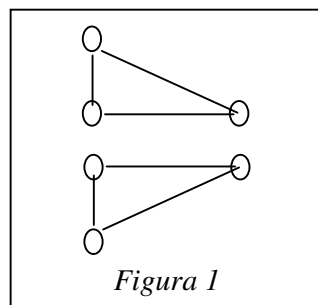
Conectamos los potenciómetros ubicados en los ejes de los segmentos a las entradas analógicas del puerto de joystick para poder medir el ángulo de un segmento respecto del otro y poder determinar la posición del punto en contacto con el objeto.

Programa de digitalización

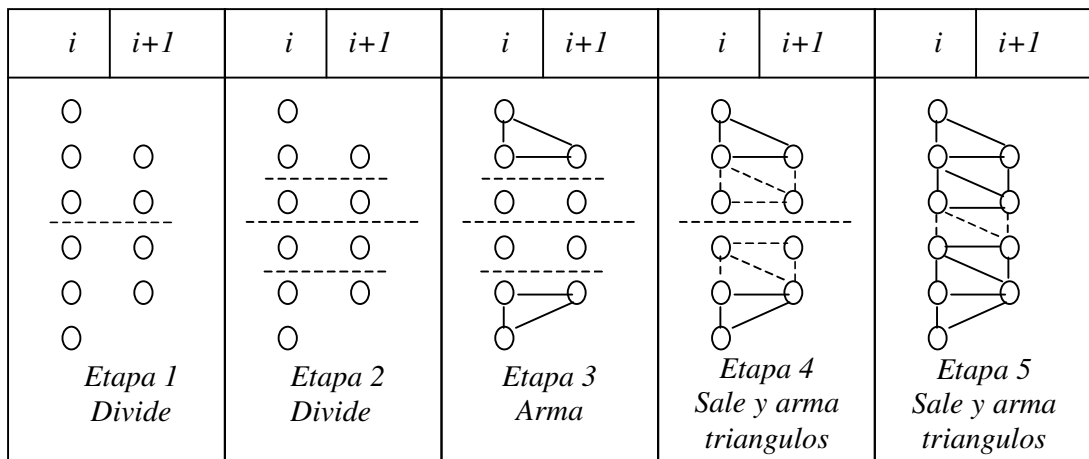
Para obtener las posiciones de los segmentos desarrollamos un programa que toma los datos del puerto y los convierte en los ángulos correspondientes.

Lo más importante es determinar la posición de la punta cuando presionamos el pulsador. Debemos conocer la medida de cada uno de los segmentos además del ángulo para poder trabajar con coordenadas polares. Las medidas son constantes y los ángulos, calculados permanentemente. Los datos son almacenados en una matriz que permite 30x72 puntos como máximo. El número 72 (columnas) se corresponde con la cantidad de ranuras que tiene la base (pues son 72 perfiles como máximo y con una separación de 5 grados) y las 30 (filas) indican la cantidad de puntos permitidos por cada perfil.

Una vez que capturamos todos los puntos, armamos los polígonos de la siguiente manera: recorriendo las columnas desde la primera hasta la anteúltima armamos triángulos con los puntos de la columna actual y la siguiente. La idea básica es ir dividiendo las columnas a la mitad en forma recursiva hasta que quede un solo punto en una de las columnas. Si en la otra columna quedan más de uno, armamos los triángulos (*figura 1*), pero si queda un punto en cada columna no se hace nada y se espera a la salida de la recursión. Repetimos el mismo procedimiento en cada una de las mitades y a la salida de la recursión unimos las mitades resultantes con dos triángulos (*figura 2*).



El siguiente ejemplo sirve para ilustrar en forma más clara el funcionamiento de este procedimiento. Tomamos las columnas i e $i+1$. Supongamos que la columna i tiene 6 puntos y la columna $i+1$ tiene 4 puntos. El procedimiento divide las columnas hasta encontrar que en una de ellas hay un solo punto. En la segunda etapa podemos ver que en la columna $i+1$ los puntos están aislados y en la columna i hay 1 y 2 puntos por cada uno de la otra columna. El paso siguiente es hacer triángulos si hay más de un punto en la columna i (*Etapas 3*). En las etapas siguientes se deben formar los triángulos a medida que va saliendo de la recursión (*Etapas 4 y 5*).



esto genera el archivo con los polígonos, el archivo de texturas y vuelve al menú principal. Si el objeto no es de revolución debemos capturar un perfil y girar la base del objeto, luego capturar el perfil siguiente y girar nuevamente la base y así sucesivamente hasta completar los 360 grados. También podemos evitar completar los 360 grados presionando la tecla ENTER, esto genera los polígonos con los puntos capturados hasta el momento.

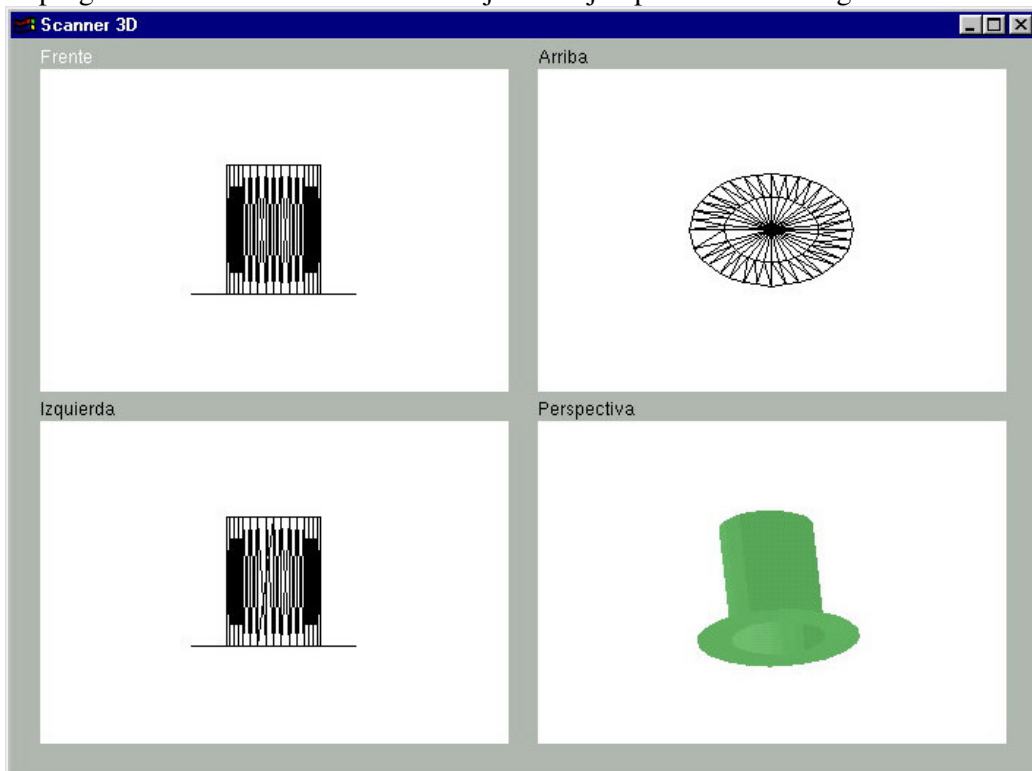
Programa de visualización

Este programa lo utilizamos para visualizar los objetos digitalizados. Para su implementación usamos el lenguaje Visual C++ y las librerías de OpenGL.

No tiene interacción con el digitalizador, pero sí entiende el formato de los archivos generados por el programa de digitalización.

El programa permite cuatro vistas del objeto. La primera es una vista de frente, la segunda, de arriba, la tercera de la izquierda y para la última ubicamos una cámara en un punto arbitrario del espacio apuntando al origen. En las tres primeras vistas observamos al objeto en estructura de alambre y en la última está coloreado con un color predeterminado o bien con alguna textura.

El programa de visualización con un objeto de ejemplo se ve de la siguiente manera:



Cuando se carga un objeto, se leen los datos de los puntos, los triángulos, la textura y el mapa de texturas, además se calculan los vectores normales por cada triángulo y se almacenan en una estructura auxiliar para no recalcularlos al momento de dibujar.

Conclusiones y trabajos futuros

En el proceso de construcción del proyecto tuvimos ciertos obstáculos que, al ser superados, nos proveyeron de nuevos conocimientos.

El principal obstáculo fue la poca precisión que brinda el puerto de joystick. Los valores de entrada varían en una o dos unidades aunque los segmentos permanezcan inmóviles, lo que se traduce en una variación del ángulo y por lo tanto varios milímetros de diferencia con el objeto real. Una posible solución futura es cambiar el puerto de entrada por uno digital y usar un conversor análogo-digital para que los valores sean exactos, pero esta solución es poco accesible en términos

económicos. Entonces optamos por mejorarlo mediante software haciendo promedios entre valores y evitar el movimiento si se sospechan oscilaciones. Esta solución nos brindó resultados aceptables debido a que los valores se mantienen fijos, el movimiento es suave y fluido y además es la más económica.

Por otra parte también tuvimos dificultades al momento de implementar algún algoritmo que arme los polígonos una vez que obtenemos los puntos. La solución fue el algoritmo que explicamos anteriormente para el armado de triángulos. La principal desventaja es que los polígonos se arman después de tomar todos los puntos y no en el momento que son obtenidos, lo que puede dar lugar a que algún triángulo no se encuentre en la posición adecuada. Pero por su forma de trabajar, dicho triángulo, no puede distar mucho de lo pretendido. Y como ventajas se pueden nombrar la simplicidad en la implementación y la uniformidad en la distribución de los polígonos. Lo más importante es que sólo debemos encargarnos de tomar los puntos individuales y no preocuparnos por indicar a que triángulo corresponde cada punto.

Los proyectos futuros relacionados con el digitalizador involucran distintos aspectos. Uno de ellos apunta a mejorar el hardware usado para la digitalización y el otro aspecto es agregar funcionalidades al programa de visualización. Estas pueden ser, por ejemplo, cortes a los objetos, cálculos de volumen, animación, etc.

Referencias bibliográficas

1. Ralf Brown, Interrupt List Release 51 1996
2. Luis Joyanes Aguilar, Programación en Turbo Pascal, serie McGraw-Hill, 1993
3. Brian W. Kernighan – Dennis M. Ritchie, El lenguaje de programación C, Prentice-Hall Hispanoamericana, 1991.
4. Addison-Wesley Publishing Company, Red Book, OpenGL Programming Guide 1996
5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes. Computer Graphics: Principles and Practice, 1990